SENG2021

Deliverable Two

Team: int elligence;

Team Members

Joel Huang (z309467), Rifa Jamal (z5311190), Eeman Chaudhry (z5309333),

Sophia Chen (z5312941), Rohan Warrier (z5312909), Rovielyn Espiritu (z5259266)

Table of Contents

Part 1: Software Architecture

External Data Source

Team int elligence; will be using the Edamam API, a free recipe database which can return recipes based on a multitude of filters such as calories, diet requirements, meal type and cuisine. Edamam contains three APIs - nutrition analysis, food database and recipe search, all of which will be useful in implementing our features.

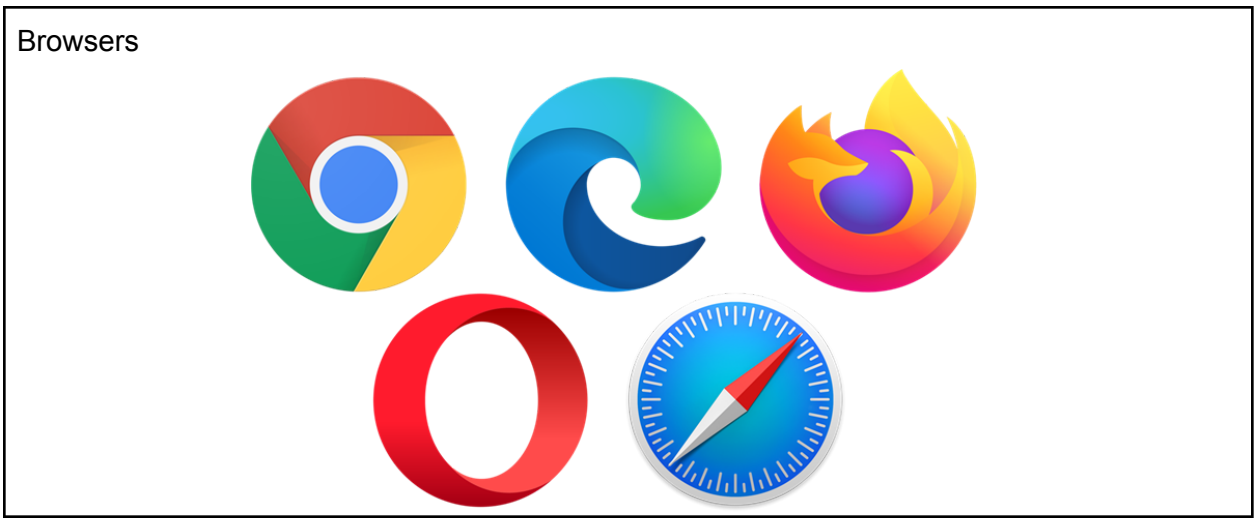
The Edamam API:

- Is time and space efficient as it has no implementation, instead specifying how to assemble software components for searching through the database and using the website. Hence, it allows us to meet objectives faster.
- Has a very structured database which can easily be filtered based on requirements, hence making the search process more efficient.
- Returns JSON which can easily be handled by Flask and python, a framework our team is experienced with.
- Is extensible as it contains many query parameters that can be easily added at any time to implement new features, for example, ‘ingr’ to specify the maximum number of ingredients the user can input.

As the API contains many search and query features, integrating it with our website allows personalisation and adaptability. For example, features can be added or removed based on developer and customer needs. The APIs query parameters also correspond to our current features and user stories.

Software Components

Web Application Stack Diagram



Front-End Framework  
Application Layer
Back-End Framework  
Database 
Deployment 

Operating System



Front-end

For our front-end client side, we will be using HTML5, CSS and Javascript with a Bootstrap Framework. These languages are easy to learn and to work with and will therefore, minimise the time needed to learn them. GitHub Pages (currently only for static hosting) will be used for front-end deployment as it allows simple hosting and changes are quick to show once refreshed.

The Bootstrap framework:

- Offers a wide range of customisable user friendly templates that is simple and easy to implement
- Requires minimal JavaScript and so is suitable for our team as we have no JavaScript or front-end web development experience
- Has a live display making it easier to edit and adapt to change
- Is flexible as it supports the major web browsers that may access our web application

Database

For our database, we will be using a Python dictionary to store the user's selected ingredients and filters. Since we only need to store a simple list that has no further relations to other data, a Python dictionary will suffice. The database will be easily accessible as we are also using Python to implement the backend of the website.

Back-end

For our back-end server side, we will be using a Python and Flask web framework as our team has previous experience using this which will minimise the learning curve when implementing our website.

Flask is suitable as it:

- Has a simple request response design that is suitable for our Edamam API which returns JSON data format.
- Is flexible as it allows you to choose which components to use and how it interacts with the rest of the code (e.g., routes), thus easy to integrate with the rest of our web stack.
- Promotes modularity due to its simplicity and fewer levels of abstractions, and therefore adaptable to changes if required
- Is more efficient as it allows unit testing due to its modularity making it easier to debug.

Python is suitable as it:

- Is a popular high-level programming language and therefore is readable, concise and easier to implement in comparison to other languages like C.
- Is open source and therefore has a wide range of libraries available making it more flexible and easier to integrate with our Edamam API and Flask
- Is an interpreted language and hence, easier to debug.

Our backend will consist of the following components:

Backend Component	HTTP method	Parameters	Return type	Descriptions
add_filters	GET	Takes in all selected filters	All recipes that match all filters	The ‘add_filters’ function runs when the user selects filters and presses the ‘find recipes’ button. The function returns all recipes that match the filters given in.
add_ingredients	GET	Takes in all included ingredients	All recipes that match ingredients user gave in	The ‘add_ingredients’ function runs when the user selects ingredients (and filters) and presses the ‘find recipes’ button. The function returns all recipes that match the ingredients given in.
display_recipes	POST	Takes in json from api	Returns the components of the recipe that need to be displayed on the webpage	Converts returned json and extracts the relevant data so that recipe can be displayed
sort_recipes	PUT	Takes in returned recipes	Returns recipes in sorted order	Sorts list of recipes according to the selected option

Platform

The machines used for the final system will require Linux as CSE uses this specifically. Therefore, everyone in our team is familiar with Linux and has access to it. Clients shall be able to access our Web-Application via any supported browser on any OS that they are using e.g. Windows or iOS.

Summary of Architectural Choices

In summary, we chose Python as our main programming language due to its flexibility, simplicity and its suitability with the rest of our Web Stack. A Flask framework will be used as it is simple and suits our API, which returns JSON data. We chose HTML, CSS and Javascript for the front-end due to its simplicity, while a Bootstrap framework provides easily customisable templates and live display. Python dictionaries are best suited for our simple database of selected ingredients and filters.

Part 2: Initial Software design

Updated User Stories

Feature: Search for recipes with specific ingredients
As a person who wants to search for recipes using a subset of ingredients
So that I can find recipes tailored to my selection
I want to be able to add ingredients I want and remove ingredients that I don't want

> **GIVEN** I am on the HomeCooks home page
> **WHEN** I click on 'Add an ingredient' button
> **THEN** I should be able to add ingredients which I want included by typing them
> **THEN** after I click on 'add filters' button
> **AND** navigate to 'Negative Search'
> **THEN** I should be able to type ingredients which I do not want included
> **WHEN** I click on 'Find recipes' button
> **THEN** I should see a list of recipes which contain most (if not all) of my selected ingredients
> **WHERE** none of the recipes include any ingredients in the 'Negative Search' list

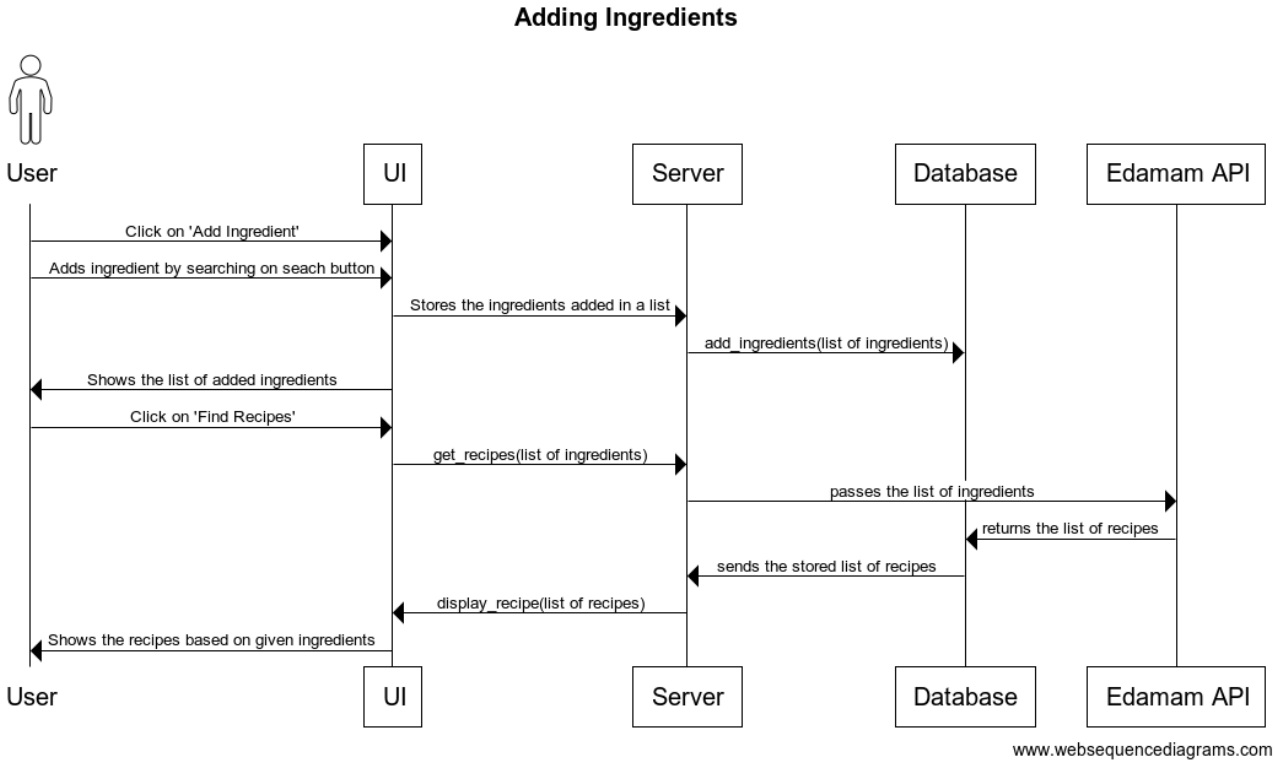
Feature: Sort recipes
As a person who wants to organise the recipe results
So that I can find recipes with a certain feature easily
I want to filter the suggested recipes by my chosen grouping

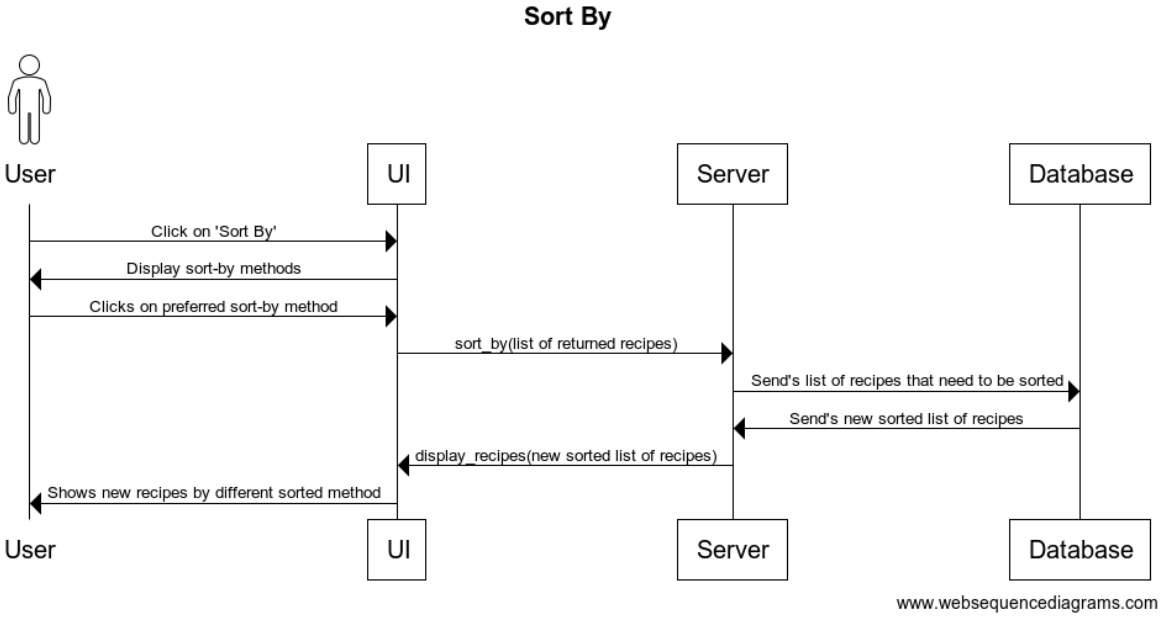
> **GIVEN** I am on the HomeCooks 'Recipe List' page
> **WHEN** I click on the 'sort by' button
> **AND** select a grouping option
> **THEN** the webpage would display recipes in the selected order

Feature: Filter recipes based on selected requirements
As a person who has specific meal requirements
So that I can only view recipes that I would be interested in cooking
I want to filter recipes based on my given constraints

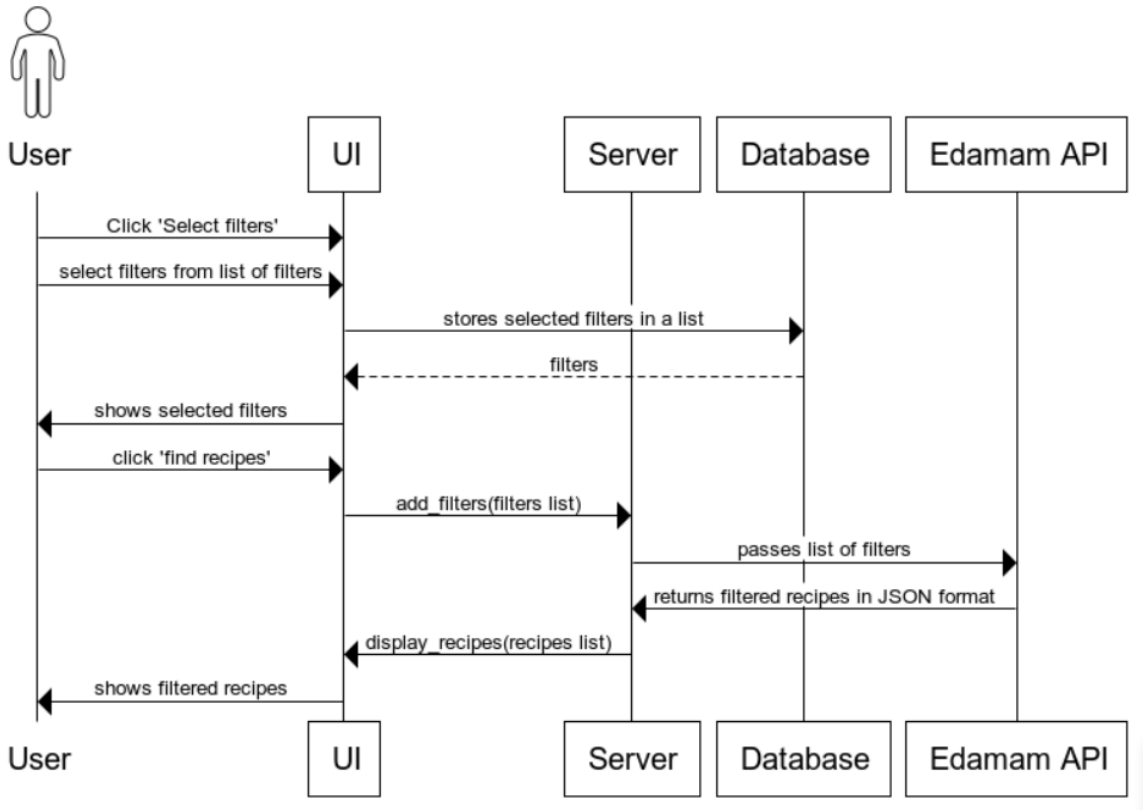
> **GIVEN** I am on the HomeCooks homepage
> **WHEN** I click on the 'select filters' button
> **THEN** I should be able to choose my desired filters
> **WHEN** I select my filters and click the 'search' button
> **THEN** the webpage will only display recipes that satisfy my selected requirements

Sequence Diagrams





Filters



Choosing a Recipe